
Alytics

GreenHouseSQL - масштабируемая система аналитики на PostgreSQL, Greenplum и Clickhouse

Max Vikharev, Alytics,
PGConf Moscow 2019

Alytics

WHO Am I

Alytics co-founder ;)

Data engineer

Software engineer

Release engineer

DevOpsDBA

Alytics

SaS система аналитики интернет маркетинга

(Система сквозной аналитики)

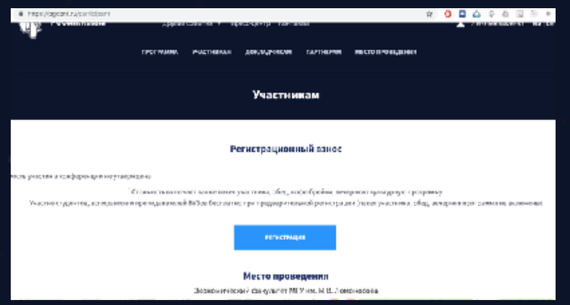
Alytics

Yandex Direct

Google AdWords

my target

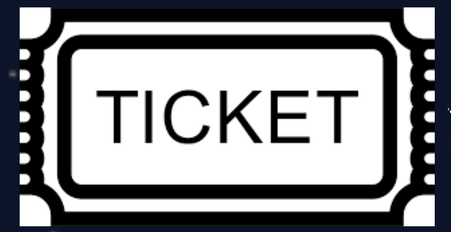
Яндекс маркет



https://pgconf.ru

Yandex Metrica

Google Analytics



Битрикс24

InSales

retail CRM

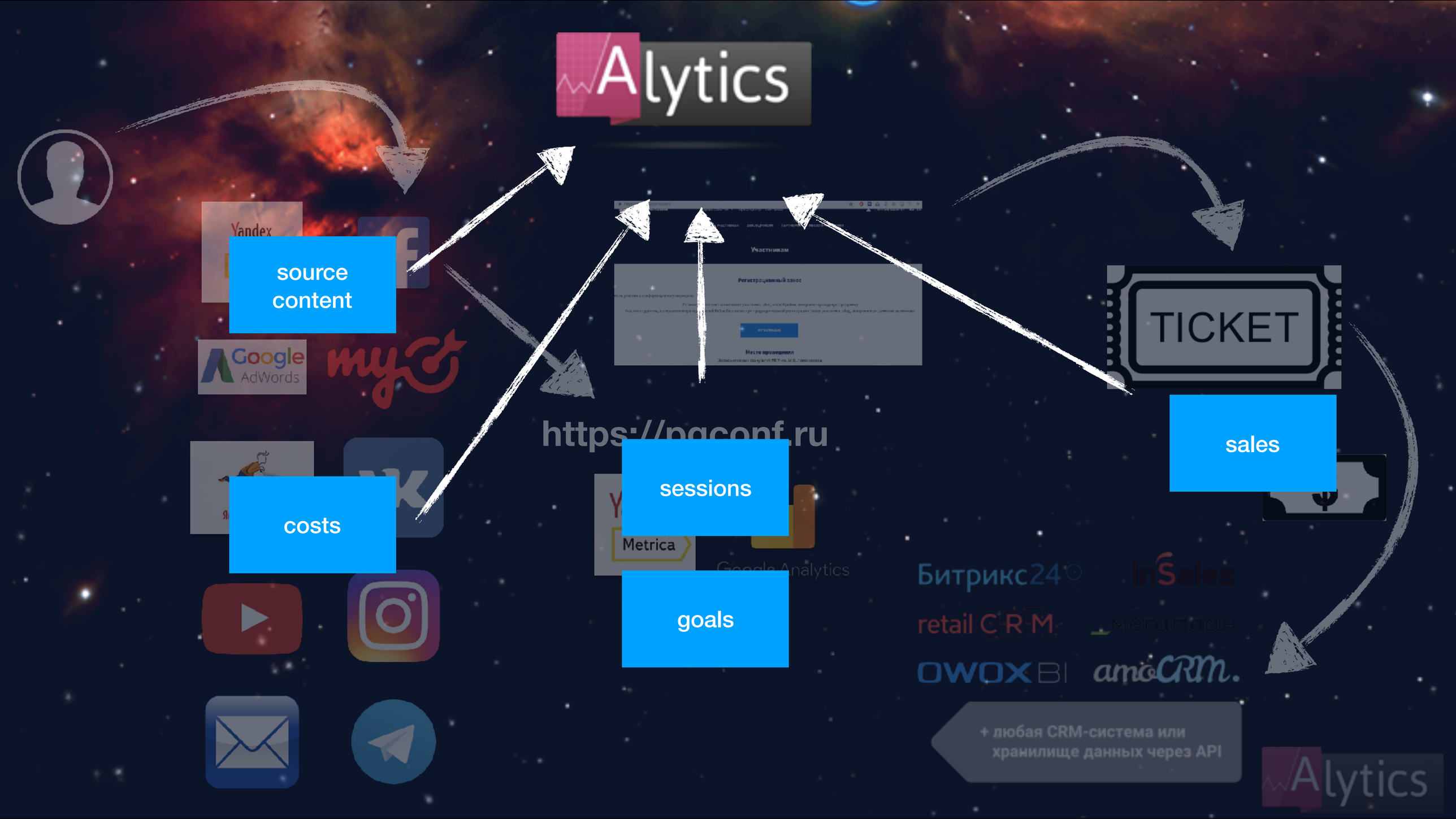
мегаплан

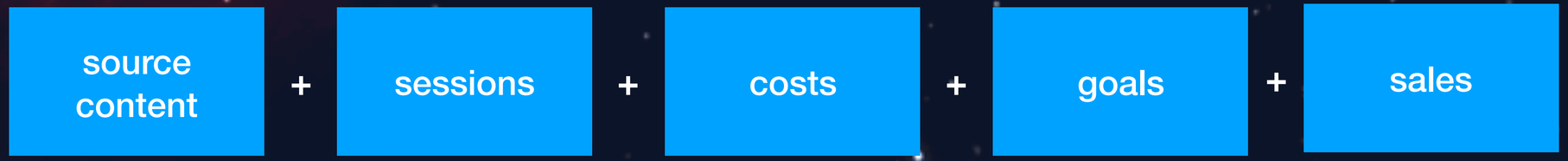
OWOX BI

amoCRM.

+ любая CRM-система или хранилище данных через API

Alytics


Alytics
source content
costs
sessions
goals
sales
TICKET
https://pgconf.ru
Участникам
Регистрационный взнос
Место проведения
Yandex
Google AdWords
Metrica
Google Analytics
Битрикс24
InSales
retail CRM
мегаплан
OWOX BI
amoCRM.
+ любая CRM-система или хранилище данных через API
Alytics



Какой трафик приносит прибыль?

Применение

Таблицы

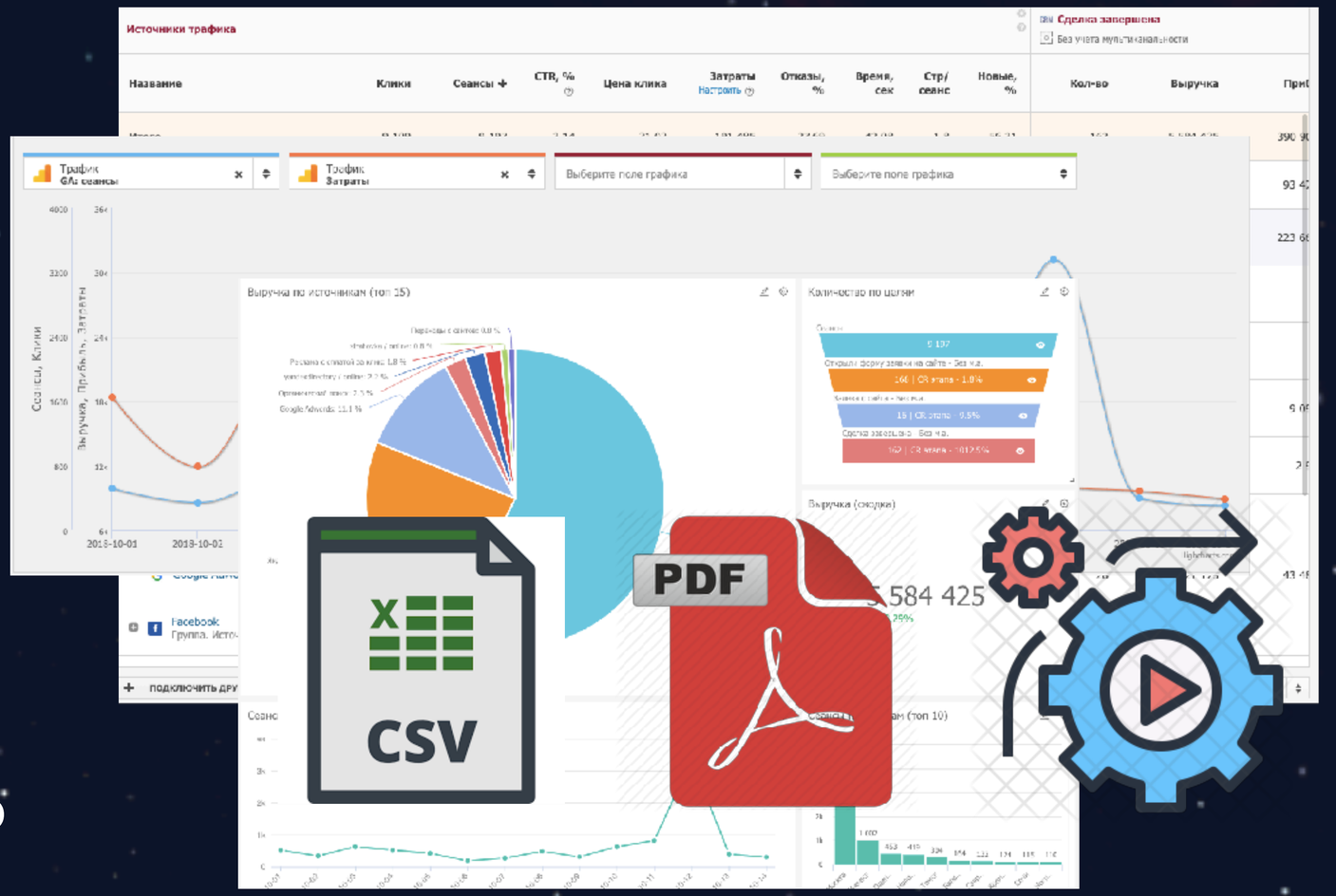
Графики

Интерактивный BI

CSV/PDF

Выгрузки по расписанию

Автоматизация



Особенности

Множество проектов - индивидуальные операции сбора данных

Регулярные ретроспективные корректировки (NOT append-only!)

Неоднородная структура фактов

Особо большие проекты

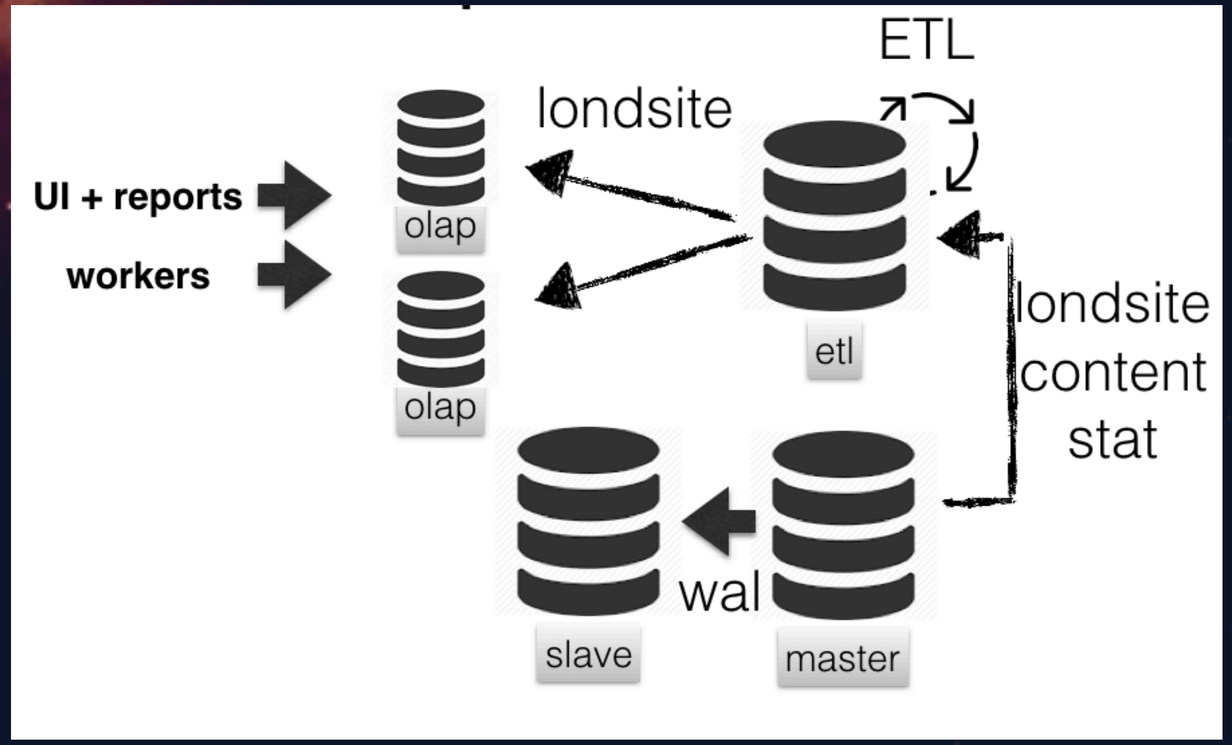
Транспонирование фактов

Непрерывное обновление контента

Синхронизации статистики три раза в день

Глубокие пересборки

Архитектура 2017



Проблемы OLAP 2017

Просадка скорости на больших проектах и глубоких периодах

Невыносимо медленные миграции схемы

Недоступные миграции данных

Сложный фейловер

Вертикальное масштабирование

Индивидуальные аггрегаты для проектов

Слабоконтролируемые синхронизации с ETL

Postgresql OLAP

Требования 2017

< 1 секунды уровень сводки за любые периоды

~ 10 секунд детальные данные включая глубокие периоды

В планах все источники трафика -> новые сервисы -> больше данных

В планах мультиканальная аналитика: сложный ETL, Windows functions. Пример.

Без раздутия стека. Python, SQL и opensource.

Идеально: обратная совместимость с старым приложением:
SQL (CTE, JOIN, Транспонирование, Множители, Сравнение периодов)

Минимальные задержки обновления контента и новых данных

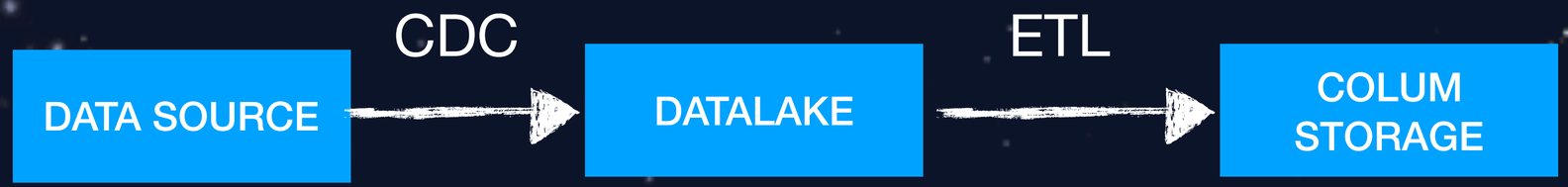
Типовая архитектура DWH решений

Datalake

Change Data Capture

ETL

Выгрука в слой колоночного хранения



Варианты на 4Q 2017

Шардинг уровня приложения (FDW, Python logic)

Postgres XL/XC

Clickhouse

pg_shard Citus

Greenplum

Greenplum

https://gpdb.docs.pivotal.io/500/best_practices/intro.html

Шардинг

Партицирование

Таблицы Heap, Appendonly, Column

Compression

ANSI SQL с оконными функциями

Высокая доступность (реплики)

MPP загрузка данных

Gp 5: Pg 8.2 -> 8.3 (json, hstore)

GP 6: Pg 9.4 (уже 9.0)

Opensource

Цикл пилотного проекта

Первоначальный сетап

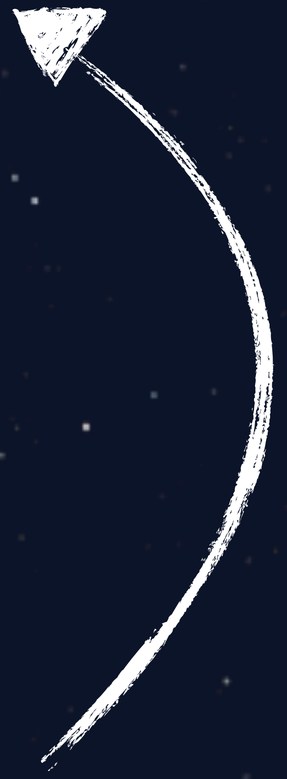
Загрузка всех данных

Тестирование производительности

Перешардирование

Доработка структуры данных

Изменение конфигурации



Загрузка
Postgresql -> Greenplum

https://gpdb.docs.pivotal.io/5160/admin_guide/external/g-external-tables.html

EXTERNAL TABLE

Script

File

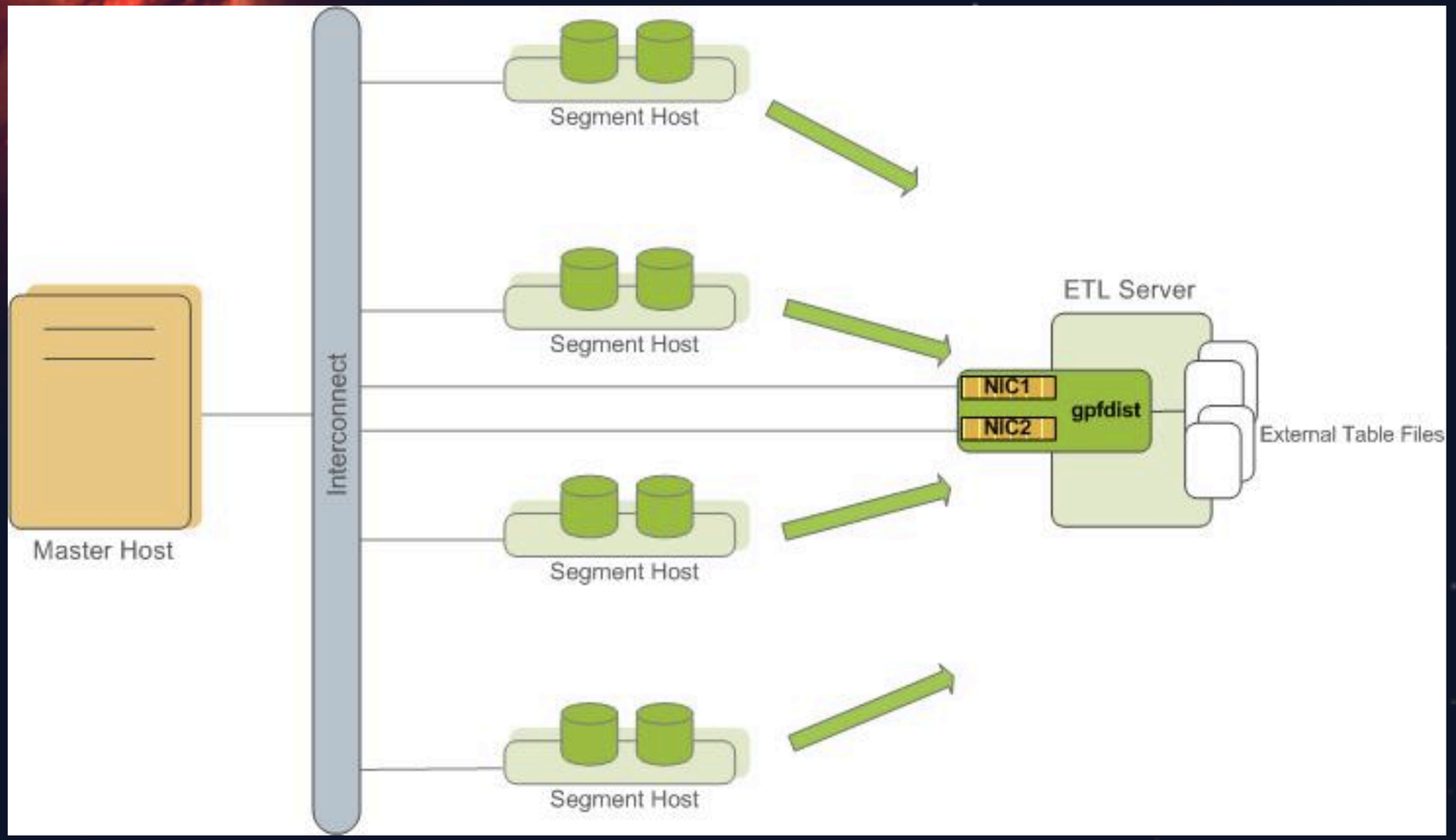
S3

Gpfdist

PXF

Gpfdist

http://engineering.pivotal.io/post/the_gpfdist_protocol_for_external_tables_in_greenplum_database/



Gpfdist

dbsource: gpfdist -p 8080 -d /gpfdist/

dbsource: psql -c “COPY (SELECT FROM table) TO ‘/gpfdist/table.csv’;”

greenplum: CREATE EXTERNAL TABLE ‘gpfdist://dbsource/table.csv';

greenplum: INSERT INTO mytable SELECT FROM external

HINT: лучше явный список полей везде (SELECT *, CREATE LIKE)

Gpfdist

dbsource: Out of space

Gpfdist

dbsource: mkfifo /gpfdist/table.csv

dbsource: psql -c “COPY (…) TO ‘/gpfdist/table.csv’;” | gzip > /gpfdist/table.csv

Факторы быстродействия

https://gpdb.docs.pivotal.io/500/best_practices/schema.html

Реляционная схема

HINT: Меньше джойнов - лучше ©, JOIN по ключу шардирования - BEST

HINT: EXPLAIN покажет дорогие REDISTRIBUTE MOTION

https://gpdb.docs.pivotal.io/500/best_practices/tuning_queries.html

Шардирование

HINT: MPP шардирование - шардирование по минимальному контенту

HINT: Перекосы в данных - зло (кейс - “)

```
select gp_segment_id, count(*) from table group by gp_segment_id;
```

Партицирование

HINT: новые партиции не наследуют параметры хранения

HINT: чем больше партиций - тем больше схема (2млн таблиц - OK)

Тип хранения

HINT: heap - часто обновляемые данные, контент

HINT: appendonly + compression - статистика

HINT: columnar - надо мерять

Индексы

HINT: Дают выхлоп, надо тестить

Изменение конфигурации

https://gpdb.docs.pivotal.io/500/best_practices/workloads.html

```
optimizer: "off"
optimizer_parallel_union: "on"
statement_mem
max_statement_mem
gp_vmem_protect_limit
gp_resqueue_priority_cpucores_per_segment
maintenance_work_mem
gp_workfile_compress_algorithm: "zlib"
shared_buffers
log_min_duration_statement
```


Автовакум выключен
… и на pg_catalog тоже
https://gpdb.docs.pivotal.io/500/best_practices/bloat.html
Alytics

Интеграция в систему. Требования

Нужно переливать исторические данные (факты) [time partition]

Нужно переливать контент (измерения) [changes]

Нужно возможность инкрементальной и полной пересинхронизации из разных БД

Интеграция в прикладные приложения с минимальными усилиями

Нужна возможность дополнительных обработок данных [ETL]

По максимуму автоматизировать operations и обновлять данные без простоя приложение

Интеграция в систему. etl-gp

Новое приложение с набором демонов, работающих в фоне

Содержит конфигурацию таблиц и баз-источников

Имеет командный интерфейс для создания таблиц и полных синхронизаций

Имеет встроенный python фреймворк для организации ETL пайплайнов

Интеграция в систему. psql-exporter

psql-exporter - микросервис интеграции psql && gpfdist

провижинится рядом с любой базой

предоставляет API для экспорта данных на файловую систему в папку gpfdist

mkfifo && COPY (custom_sql) | gzip > file ; DELETE file

Взаимодействие etl-gp - psql-exporter

Делпоим psql-exporter на БД источник

```
root@db-olap-03 /var/docker/gpfdist/alytics_olap_01 # cat docker-compose.yml
version: '2'

services:

  gpfdist:
    image: dr.alytics.org/greenplum-loaders:5.1.0
    command: /bin/bash -c "source greenplum_loaders_path.sh; gpfdist -p 7070 -t 600 -d /gpfdist_data/ -m 655350"
    restart: always
    network_mode: host
    volumes:
      - /var/www/gpfdist/alytics_olap_01:/gpfdist_data/
      - /etc/localtime:/etc/localtime

  psql-data-exporter:
    image: dr.alytics.org/psql-exporter:master
    restart: always
    network_mode: host
    environment:
      - EXPORT_DIR=/gpfdist_data/
      - PORT=7071
      - PG_HOST=
      - PG_PORT=54351
    volumes:
      - /var/www/gpfdist/alytics_olap_01:/gpfdist_data/
      - /etc/localtime:/etc/localtime
```

Alytics

Взаимодействие etl-gp - psql-exporter

Делпоим psql-exporter на БД источник

Описываем конфигурацию БД источников в etl-gp

```
DATABASES = {
    'stat': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': os.getenv('STAT_DB_DB'),
        'USER': os.getenv('STAT_DB_USER'),
        'PASSWORD': os.getenv('STAT_DB_PASSWORD'),
        'HOST': os.getenv('STAT_DB_HOST'),
        'PORT': os.getenv('STAT_DB_PORT'),
        'GPFDIST_HOST': os.getenv('STAT_DB_GPFDIST_HOST'),
        'GPFDIST_PORT': os.getenv('STAT_DB_GPFDIST_PORT'),
        'PSQL_EXPORTER_HOST': os.environ.get('STAT_DB_GPFDIST_EXPORTER_HOST'),
        'PSQL_EXPORTER_PORT': os.environ.get('STAT_DB_GPFDIST_EXPORTER_PORT'),
        'TEST': {
            'MIRROR': 'default'
        }
    },
```

Взаимодействие etl-gp - psql-exporter

Делпоим psql-exporter на БД источник

Описываем конфигурацию БД источников

Описываем конфигурацию таблиц

```
class AnalyticsStatPremium(SimpleTable):
    source_db = "stat"
    table_name = "analytics_stat_premium"
    fields = {
        "time_on_site": "integer",
        "access_id": "integer",
        "pageviews": "integer",
        "new_visits": "integer",
        "date": "date",
        "content_id": "integer",
        "bounces": "integer",
        "id": "integer",
        "visits": "integer",
    }
    parameters = "appendonly=true, compresslevel=7"
    distribution = "content_id"
    partitioning = "RANGE (date) ( START (date '201
                   "INCLUSIVE END (date '2018-10-01
                   "EXCLUSIVE EVERY (INTERVAL '1 wee
```

Alytics

Взаимодействие etl-gp - psql-exporter

Делпоим psql-exporter на БД источник

Описываем конфигурацию БД источников

Описываем конфигурацию таблиц

Запускаем команды полной синхронизации

```
./manage.py resync_stat --tables=analytics_sessions_stat
```

Alytics

Отслеживание изменений контента.

Триггерный SQL механизм отслеживания изменений

Устанавливается командой etl-gp в базу источник

Фоновый процесс в цикле переливает данные

Отслеживание изменений time partition

Воркеры приложений после завершения обновления данных по проекту репортят в etl-gp информацию о чанках (project_id, date)

Демон etl-gp сохраняет в мета-бд информацию о чанках

Фоновый процесс в цикле переливает данные по накопившимся чанкам

ETL процессы

После обновления любой из таблиц можно запустить ETL процесс

Соглашение:

Public schema - для сырых данных (Datalake)

Service schema - для результатов ETL (подготовленные данные для работы из прикладного приложения)

Один ETL процесс обработки на таблицу

Обновление без простоя

ETLM*

*M - maintenence

Worker: stat_table updated

Chunks: (project_id1, date1), …, (project_idN, dateN) -> POST etp-gp/chunks/

EtI-gp: Chunks -> SQL: SELECT FROM stat_table WHERE project_id1, date1 -> psql-exporter

CREATE EXTERNAL TABLE external_table … ‘gpfdist://';

DELETE + INSERT INTO public.stat_table FROM external_table

Detect partition by changes dates

```
sql = """
    SELECT DISTINCT partitiontablename, partitionrank, partitionrangestart, partitionrangeend
    FROM (
            SELECT date_trunc('day', date)::TEXT AS x
            FROM generate_series
            ( '{date_from}'::TIMESTAMP
            , '{date_to}'::TIMESTAMP
            , '1 day'::INTERVAL) date
        ) d,
        pg_partitions p
    WHERE p.tablename = '{table}'
        AND p.partitionschemaname = '{schema}'
        AND d.x::date >= split_part(p.partitionrangestart, '::', 1)::date
        AND d.x::date < split_part(p.partitionrangeend, '::', 1)::date;
"""
```

Обновление без простоя

ETLM*

*M - maintenence

Worker: stat_table updated

Chunks: (project_id1, date1), …, (project_idN, dateN) -> POST etp-gp/chunks/

EtI-gp: Chunks -> SQL: SELECT FROM stat_table WHERE project_id1, date1 -> psql-exporter

CREATE EXTERNAL TABLE external_table … ‘gpfdist://';

DELETE + INSERT INTO public.stat_table FROM external_table

Detect partition by changes dates

VACUUM ANALYZE public.stat_table_partition_w2019p1

Transform (public.stat_table_partition_w2019p1) -> service. stat_table_for_exchange

ANALYZE service.stat_table_for_exchange

EXCHANGE PARTITION service.stat_table_for_exchange -> service.stat_table_partition_w2019p1

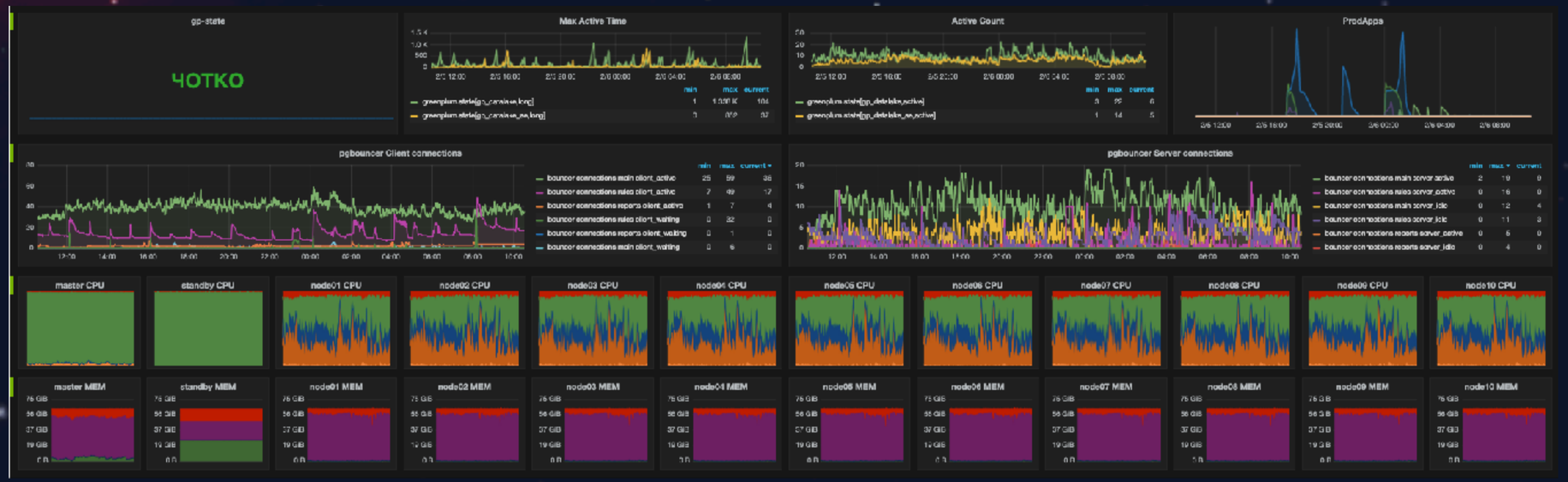
ANALYZE ROOTPARTITION service.stat_table

Application -> service.stat_table

Мониторинг и контроль

Мониторинг и контроль

Доступость Greenplum - gpstate



Фейловер

Восстанавливаем

```
gprecoverseg -a
```

Если не помогло

```
gprecoverseg -aF
```

Проверяем состояние

Ребалансировка

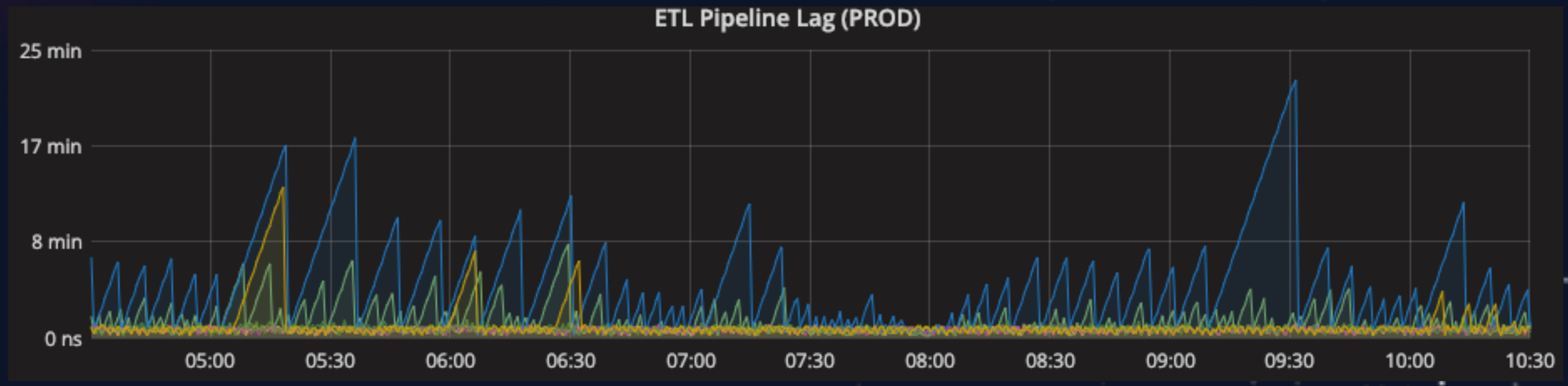
```
gprecoverseg -r
```

Проверяем состояние

Мониторинг и контроль

Доступость Greenplum - gpstate

Лаги переливок и процессов ETL (пайплайнов) - grafana



Мониторинг и контроль

Доступость Greenplum - gpstate

Лаги переливок и процессов ETL (пайплайнов) - grafana

Per-database health check, https://etl-gp.prod.cluster.local/health

Админка чанков

API готовности пайплайнов и индивидуальных чанков

Профит

Все по 10 секунд без аггрегатов

Сокращение ночного ворклода с 7-9 часов до 1 - 1,5

500 глубоких TPCH с 12 часов до 30 минут

Уменьшение размера кластера в 7-10 раз

gprecoverseg - восстановление в одну команду

https://gpdb.docs.pivotal.io/500/best_practices/ha.html

Мифы о железе

2 x TOP DELL -> 14 cheap noname

Сеть 1Гбит

SSD RAID 0

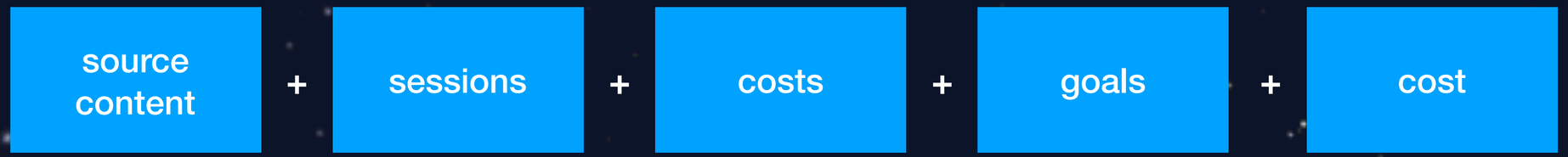
Недостатки

Все по 10 секунд

Быстрый слой

плоские таблицы

подготовка слоя данных для быстрых выборок



Проблемы для GP

Глобальный планировщик

Транспонирование фактов

Clickhouse

Встроенное шардирование

Планировщик заточен под локальные данные

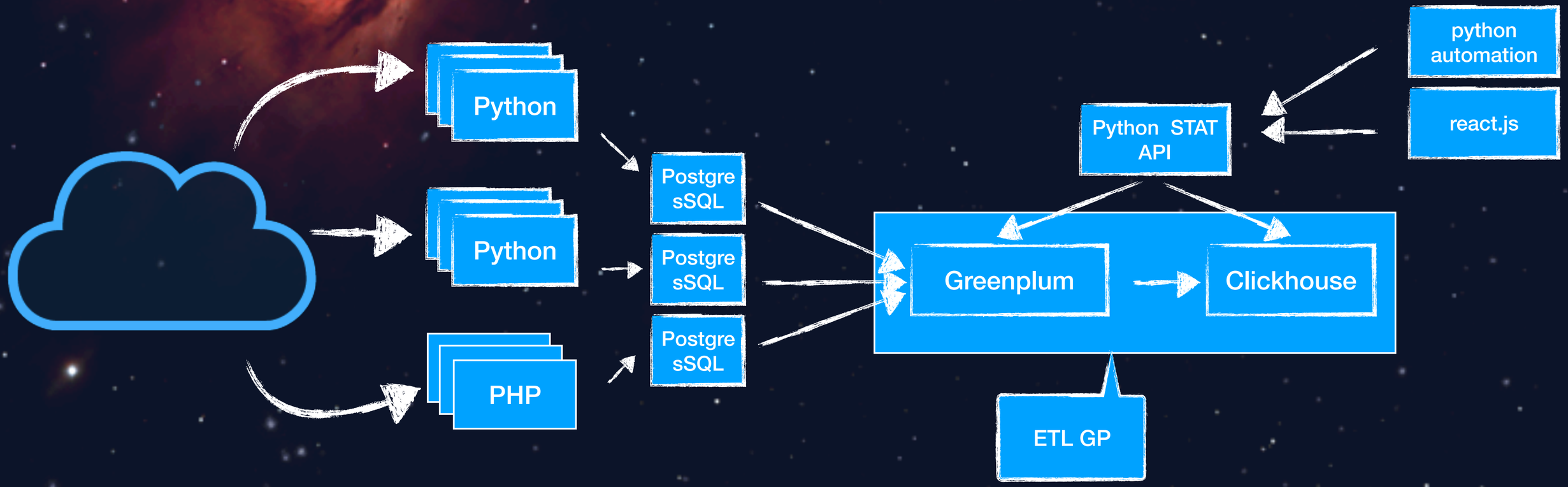
Встроенные вложенные структуры Goal

Мультимастер

Merge движки таблиц для обновлений в фоне

Любые запросы ~ 10*N мс

Общая архитектура



Профит

10 секунд детальные данные включая глубокие периоды

Менее 1 секунды уровень сводки и BI

Быстрый MPP ETL слой с ANSI SQL и оконными функциями

In-memory быстрый слой данных

Без переписывания приложений

Задержка контента - минуты Задержка статы - до часа*

Alytics

Спасибо

Вопросы?

Max Vikharev
mvikharev@alytics.ru
@mvikharev

Opensource для psql-exporter && etl-gp: https://alytics.ru/pg-gp-to-opensource

Группа greenplum [RUSSIAN COMUNITY]: https://t.me/joinchat/DYEBbRGynrZ0dL-QaKmkbw

Группа clickhouse [RUSSIAN COMUNITY]: @clickhouse_ru

https://www.meetup.com/Scale-out-databases-and-engines/events/257153368/

Alytics